Stochastic Gradient Descent for Numerical MAX-CSP

Derek Paulsen

1 Abstract

2 Introduction

3 Background

Constraint satisfaction problems (CSPs) encompasses many different problems. Formally a constraint satisfaction problem (CSP) is defined as a triple $\langle X, D, C \rangle$, where

$$\begin{aligned} X &= \{X_1, ..., X_n\} \text{ the set of variables} \\ D &= \{D_1, ..., D_n\} \text{ the domains of each respective variable} \\ C &= \{C_1, ..., C_m\} \text{ the set of constraints} \end{aligned}$$

A classic example of CSP, is the 3-SAT problem. In this case the boolean variables that appear in at least one clause would be X. The domain for each variable would be $D_i = \{true, false\}$ and the constraints would all be three variable disjunctive clauses. The 3-SAT problem of course known to be NP-Complete. In this paper we focus on a subset of CSPs called Numerical MAX-CSP. In typical a CSP all constraints must be satisfied, however in MAX-CSP the goal is simply to satisfy as may constraints as possible, additionally the domains are the variables are numerical (e.g. the real numbers).
While this problem has certainly been considered before, we were only able to find one previous work that directly address this issue.
In this paper the authors propose a solution to numerical MAX-CSP. In particular where $D_i = \mathbb{R}$ and each constraint $C_j = a_j^T x \leq b_j, a_j \in \mathbb{R}^m, b_j \in \mathbb{R}$. That is, given a set of linear inequalities, satisfy as many as possible. In the paper the authors propose a specific algorithm for this problem based on branch and bound techniques. Unfortunately, this algorithm is worst case exponential time. In fact, this problem is easy to formulate as a mixed integer program,

$$\begin{aligned} &\min_{x,z} \quad 1^T z \\ s.t. \quad & Ax - \epsilon z \leq 0 \\ & w \in \mathbb{R}^n \\ & z \in \{0,1\}^m \end{aligned}$$

Where ϵ is some large number. The general problem of mixed integer linear programming is of course NP-Hard in general (by reduction to 0-1 integer programming which is NP-Complete). Hence this numerical MAX-CSP is likely not to admit an efficient solution.

4 Motivation

Numerical MAX-CSPs have a wide variety to applications in such as debugging infeasible linear programs, however we are interested in one particular problem, which is that of tuning the weights for full search engines.
Full text search engines are used in a plethora of applications. These search engines (such as Apache Lucene) are built for efficient retrieval of top-k documents based on a TF/IDF based scoring metric which is dot product between two sparse vectors $q^T d = score$[[1]][1][[2]][2][[3]][3]. While this default scoring gives decent results

out of the box, it is frequently augmented by re-weighting the query q, with some weight vector w changing to scoring to be $(q \odot w)^T d = score$. This allows for boosting of certain terms or fields to improve the quality of search results while not changing the underlying search algorithm.
The problem we will address in this project is finding a good weight vector w. In particular our problem setting is as follows. We are giving a set of query vectors $Q = \{q_1, ...q_n\}$. For each of these query vectors q_i we are given a set of k retrieved document vectors with labels $R_i = \{(d_{i,1}, l_{i,1}), ..., (d_{i,k}, l_{i,k})\}$, $l \in \{relevant, irrelevant\}$. We wish to find a weight vector w that minimizes the number of irrelevant documents that are examined before the relevant result is found. This problem can be formulated as a mixed integer linear program (MILP) of the following form

$$\begin{aligned} \min_{w,z} \quad & 1^T z \\ s.t. \quad & Aw - \epsilon z \leq 0 \\ & w \geq 0 \\ & z \in \{0, 1\} \end{aligned}$$

5 Problem

6 Our Solution

7 Experiments

8 Discussion

9 Conclusions